Elsevier Editorial System(tm) for Computers and Fluids
Manuscript Draft

Manuscript Number:

Title: Recent ship hydrodynamics developments in the parallel two-fluid flow solver Alya

Article Type: Special Issue: ParCFD2011

Keywords: Incompressible two-phase flows; stabilized finite element methods; level set; ship hydrodynamics

Corresponding Author: Dr. Herbert Owen,

Corresponding Author's Institution: Barcelona Supercomputing Center (BSC-CNS)

First Author: Herbert Owen

Order of Authors: Herbert Owen; Guillaume Houzeaux; Cristobal Samaniego; Anne Cecile Lesage; Mariano Vázquez

Abstract: CFD modeling of turbulent free surface flows has become an important tool in the design of ship hulls. A two-fluid flow solver that can predict the flow pattern, free surface shape and the forces on the ship hull is presented. The discretization is based on unstructured linear finite elements, tetrahedras and prisms. A Variational Multiscale Stabilization technique known as Algebraic Sub Grid Scale Stabilization (ASGS) is used to deal with convection dominated flows and allow for equal order velocity-pressure
interpolations. A fixed grid method that captures the position of the interface with the Level Set technique is used to simulate the two-phase flow. The jump in the fluid properties is smoothed in a region close to the interface. Spalart Allmaras and SST k-ω turbulence models have been tested without showing significant differences. The need to accurately predict the viscous forces on the ship hull has motivated the use of anisotropically refined prism elements close the hull. Such meshes have challenged the
capabilities of the equation based reinitialization techniques that we had previously used for the Level Set method.

Recent ship hydrodynamics developments in the parallel two–fluid flow solver Alya

H. Owen[a,*], G. Houzeaux[a], C. Samaniego[a], A.C. Lesage[a], M. Vázquez[a]

[a] *Barcelona Supercomputing Center (BSC-CNS), Edificio NEXUS I, Campus Nord UPC, Gran Capitán 2-4, 08034 Barcelona, Spain.*

Abstract

CFD modeling of turbulent free surface flows has become an important tool in the design of ship hulls. A two-fluid flow solver that can predict the flow pattern, free surface shape and the forces on the ship hull is presented. The discretization is based on unstructured linear finite elements, tetrahedras and prisms. A Variational Multiscale Stabilization technique known as Algebraic Sub Grid Scale Stabilization (ASGS) is used to deal with convection dominated flows and allow for equal order velocity-pressure interpolations. A fixed grid method that captures the position of the interface with the Level Set technique is used to simulate the two-phase flow. The jump in the fluid properties is smoothed in a region close to the interface. Spalart Allmaras and SST k–ω turbulence models have been tested without showing significant differences. The need to accurately predict the viscous forces on the ship hull has motivated the use of anisotropically refined prism elements close the hull. Such meshes have challenged the capabilities of the equation based reinitialization techniques that we had previously used for the Level Set method.

Keywords: Incompressible two–phase flows, stabilized finite element methods, level set, ship hydrodynamics

1. Introduction

Ship design currently depends more heavily on the CFD modeling of turbulent free surface flows than on tank and wind tunnel testing as indicated by the results of the two most competitive boat races, America's Cup and Volvo Ocean Race [1]. Simulations are cheaper, faster and more reliable than traditional test for boat design. Moreover, thanks to the constant increase in computer memory and speed, together with improvements in the numerical algorithms, the costs are constantly decreasing. Simulations are run at full scale avoiding errors due to scaled test. If scaled experimental results exist, running simulation with both the scaled and real sizes can indicate errors due to the scaling process. Enhanced flow visualization and force decomposition, provides much richer information than the one measurable in tank tests leading to a much better understanding of the flow phenomena.

CFD approaches for free surface and two-fluid problems can be categorized into two main groups: Fixed mesh interface capturing techniques and moving mesh interface tracking techniques. For complex problems the interface capturing approach is usually preferred. Alya code uses a Level Set technique [2, 3] to capture the position of the interface. Fixed mesh methods generally share two basic steps, one where the motion in both phases is found as the solution of the Navier–Stokes equations with variable properties and the other one, where an equation for an interface function that allows to determine the position of the interface, and thus the properties to be assigned in the Navier–Stokes step, is solved.

The numerical formulation presented here to solve the incompressible Navier–Stokes equations uses a time discretization based on the standard trapezoidal rule and a stabilized finite element method referred to as Algebraic Sub-Grid Scales (ASGS) [4]. It is designed to allow both *equal velocity–pressure interpolations* (thus avoiding the need to satisfy the classical inf-sup condition) and to deal with *convection–dominated flows*, that is, situations in which the cell Reynolds number is greater than unity.

The Level Set method leads to a transport partial differential equation the solution of which determines the position of the free surface as an isovalue of the unknown of this equation, which we will call ψ. This equation is hyperbolic and therefore it is also necessary to use a stabilized finite element method to solve it. The proper functioning of the Level Set technique requires that the module of the gradient of the Level Set function is close to one. As the flow evolves it deviates from this value and therefore the Level Set function needs to be reinitialized. The reinitialization of the Level Set function consists in finding a new Level Set function with module of its gradient close to one and that minimizes the displacement of the interface during the reinitialization. In ship hydrodynamics the need for reinitialization is particularly noticeable in the bow and stern. The use of anisotropic refined meshes close to the ship hull to accurately model the viscous forces poses important difficulties to differential equation based reinitialization techniques.

Accurate viscous solutions require refined meshes close

[*] Principal corresponding author
[**] Corresponding author
Email address: herbert.owen@gmail.com. Phone number: +34934054290 (H. Owen)

to the wall. In order to simulate flows at high Reynolds numbers the use of turbulence models is mandatory. In ship hydrodynamics, the usual approach is to use a family of models known as Reynolds averaged Navier Stokes (RANS). The two most popular approaches are Spalart Allmaras [5] and SST k-ω [6]. Our numerical experiments indicate that for the problems we have tested there is no significant difference between the results they provide. For the examples presented in this paper we use SST k-ω.

The remainder of the paper is organized as follows. In Section 2 we describe the mathematical model used to solve the Navier–Stokes equations and in Section 3 we briefly describe the Level Set Method used. Finally in Section 4 we present some numerical examples.

2. The two-fluid Navier Stokes equations

The velocity and pressure fields of two incompressible fluids moving in the domain $\Omega = \Omega_1 \cup \Omega_2$ during the time interval (t_0, t_f) can be described by the incompressible two–fluid Navier–Stokes equations [7]:

$$\rho\Big[\partial_t \mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u}\Big] - \nabla\cdot(-p\mathbf{I} + 2\mu\varepsilon(\mathbf{u})) = \mathbf{f}, \qquad (1)$$

$$\nabla\cdot\mathbf{u} = 0, \qquad (2)$$

where ρ is the density, $\mathbf{u}$ the velocity field, μ the dynamic viscosity, p the pressure, $\varepsilon(\cdot)$ the symmetric gradient operator and $\mathbf{f}$ the vector external body forces, which includes the gravity force $\rho\mathbf{g}$ and buoyancy forces, if required. The density, velocity, dynamic viscosity and pressure are defined as

$$\mathbf{u},\ p,\ \rho,\ \mu = \begin{cases} \mathbf{u}_1\ \ p_1,\ \ \rho_1,\ \ \mu_1 & \mathbf{x}\in\Omega_1, \\ \mathbf{u}_2\ \ p_2,\ \ \rho_2,\ \ \mu_2 & \mathbf{x}\in\Omega_2, \end{cases}$$

where Ω_1 indicates the part of Ω occupied by fluid number 1 and Ω_2 indicates the part of Ω occupied by fluid number 2. The extent of Ω_1 and Ω_2 is given by the level set function ψ. To ensure stability the density and viscosity jumps are smoothed at the interface [8] using a smoothed Heaviside function.

Let σ be the stress tensor and $\mathbf{n}$ the unit outward normal to the boundary $\partial\Omega$. Denoting by an over-bar prescribed values, the boundary conditions to be considered are:

$$\begin{array}{rl} \mathbf{u} = \overline{\mathbf{u}} & \text{on } \Gamma_{\rm du}\times(t_0,t_f), \\ \mathbf{n}\cdot\sigma = \mathbf{t}_{\rm nu} & \text{on } \Gamma_{\rm nu}\times(t_0,t_f), \\ \mathbf{u}\cdot\mathbf{n} = 0, \quad \mathbf{n}\cdot\sigma - (\mathbf{n}\cdot\sigma\cdot\mathbf{n})\,\mathbf{n} = \mathbf{t}_{\rm mu} & \text{on } \Gamma_{\rm mu}\times(t_0,t_f). \end{array} \qquad (3)$$

Observe that $\Gamma_{\rm du}$ is the part of the boundary with Dirichlet velocity conditions, $\Gamma_{\rm nu}$ the part with Neumann conditions (prescribed stress) and $\Gamma_{\rm mu}$ the part with mixed conditions. These three parts do not intersect and are a partition of the whole boundary $\partial\Omega$. $\mathbf{t}_{\rm nu}$ is the traction on $\Gamma_{\rm nu}$ and $\mathbf{t}_{\rm mu}$ is the tangential component of the traction on $\Gamma_{\rm mu}$. In this work $\mathbf{t}_{\rm mu}$ is calculated with a Reichardt's wall law [9]. Initial conditions have to be appended to the problem.

Let ρ_0 be the initial density for a flat free surface and $\mathbf{f} = \rho\ \mathbf{g}$. We can subtract $\rho_0\ \mathbf{g}$ from both sides of equation (1) and suppose gravity is in the z direction, $\mathbf{g} = g\mathbf{e}_z$, to obtain

$$\rho\Big[\partial_t \mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u}\Big] - \nabla\cdot(2\mu\varepsilon(\mathbf{u})) + \nabla(p - p_0) = (\rho - \rho_0)\,\mathbf{g},$$

where p_0 is the initial hydrostatic pressure calculated as

$$p_0 = \int_{z_{fs0}}^{z} \rho_0\ g\ dz,$$

where z_{fs0} is the z coordinate of the initial flat free surface and z is the z coordinate of any point in the domain. Now we can define a new pressure, $p^* = p - p_0$, and a corresponding stress tensor $\sigma^* = -p^*\mathbf{I} + 2\mu\varepsilon(\mathbf{u})$. This is the approach we use in our code and the pressure we have plotted for in the numerical examples is p^*. The boundary condition at the outlet in the normal direction is then simply $\mathbf{n}\cdot\sigma^*\cdot\mathbf{n} = 0$. At the continuous level this is equivalent to prescribing the normal traction, $\mathbf{n}\cdot\sigma\cdot\mathbf{n}$, equal to the initial hydrostatic pressure.

The equations are discretized in space with the finite element method. ASGS [4] stabilization is used to deal with convection-dominated flows and to circumvent the well known div-stability restriction for the velocity and pressure finite element spaces [10], allowing in particular equal interpolation for both unknowns. For the time discretization the generalized trapezoidal rule is used and the convective term is linearized using a Picard scheme.

A straightforward way to solve the discretized Navier-Stokes equations is to consider the monolithic scheme, that is to solve the momentum and continuity equations in a coupled way. This approach can lead to badly conditioned systems that would require very robust preconditioners such as ILU that have bad speedup properties, precluding their use on large scale computers. The most common approach to avoid this bad conditioning is to use fractional steps techniques [11, 12, 13, 14, 15] that uncouple velocity and pressure but introduce errors due the splitting. In this work we use an Orthomin(1) solver [16] that converges to the monolithic scheme and uses an approximate Schur complement preconditioner similar to fractional step schemes, thus leading to similarly well conditioned systems [17, 18]. It has proved to be sequentially efficient and capable of scaling up to thousands of processors.

3. Implementation of the level set method

The basic idea of the level set method is to define a smooth scalar function, say $\psi(\boldsymbol{x}, t)$, over the computational domain Ω that determines the extent of subdomains Ω_1 and Ω_2. For instance, we may assign positive values to the points belonging to Ω_1 and negative values to the points belonging to Ω_2. The position of the fluid front will be defined by the iso-value contour $\psi(\boldsymbol{x}, t) = 0$. The evolution of the front $\psi = 0$ in any control volume $V_t \subset \Omega$ which is moving with a divergence free velocity field $\boldsymbol{u}$ leads to:

$$\partial_t \psi + (\mathbf{u} \cdot \nabla)\psi = 0 \tag{4}$$

Function ψ is the solution of the hyperbolic equation (4) with the boundary conditions:

$$\psi = \overline{\psi} \qquad \text{on } \Gamma_{\text{inf}} \times (t_0, t_f),$$

$$\psi(\mathbf{x}, 0) = \psi_0(\mathbf{x})$$

The initial condition ψ_0 is chosen in order to define the initial position of the fluid front to be analyzed. The boundary condition $\overline{\psi}$ determines which fluid enters through a certain point of the inflow boundary.

Due to the pure convective type of the equation for ψ, we use the SUPG technique for the spatial discretization. Again, the temporal evolution is treated via the standard trapezoidal rule.

For the numerical solution of the level set equation it is preferable to have a function without large gradients. Since the only requirement such a function must meet is $\psi = 0$ at the interface, a signed distance function $(\mid \nabla\psi \mid \ = 1)$ is used. Under the evolution of the level set equation, ψ will not remain a signed distance function and thus needs to be reinitialized. This can be achieved by redefining ψ for each node of the finite element mesh according to the following expression:

$$\psi = \operatorname{sgn}(\psi^0)d$$

where ψ^0 stands for the calculated value of ψ, d is the distance from the node under consideration to the front, and $\operatorname{sgn}(\cdot)$ is the signum of the value enclosed in the parenthesis.

In Figure 1 we show the level set function on the symmetry plane of a race boat hull together with the velocity field close to the bow and stern. Close to the bow the Level Set contour lines close to the zero contour are spread out as they are transported by the flow leading to low Level Set gradients. On the contrary, close to the stern the Level Set contour lines are clustered by the flow leading to high gradients. Both situations are unfavorable and these are the two regions were reinitialization is needed most.

In order to calculate the distance d we are currently using a geometrical method based on a skd-tree [19]. Computing the distance from a point to a surface mesh is a crucial issue in the implementation of the level set reinitialization. For each point where one wants to know the distance, it involves searching among all the triangular faces into which the surface is divided the one that gives the minimum distance. This search can affect the performance of the whole system in a negative way: the time of simulation can grow considerably. To reduce the number of computations of this expensive task, we use geometric search structures to define the surface mesh. Our implementation uses a bounding volume hierarchy known as skd-tree. In these binary trees each node has a set of faces of the surface mesh and their corresponding associated bounding volume. The implementation details for building these structures are described in [19].

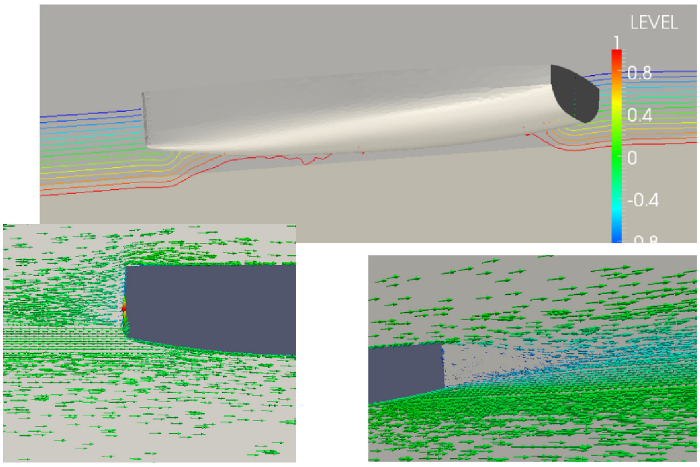

Figure 1: Need for reinitialization on a race boat hull.

Reference [19] also describes a way to use the skd-trees to determine the distance between a point and the surface mesh. The idea is to minimize an upper bound on the distance between the point and the surface mesh while traversing the binary tree from the root node. This upper bound is computed as the distance to the farthest point in the bounding volume corresponding to the current node.

In our experience, better results are obtained if we first visit the nodes whose bounding volumes are closest to the point when we traverse the tree from the root node. In our implementation, no list of candidate nodes is used.

We have also tested differential equation based reinitialization [2, 3]. Differential equation is efficient and has good scalability. We have found it works fine with isotropic meshes without boundary layer prism elements. Unfortunately it shows poor robustness on anisotropic meshes, such as the ones used in this work. This is the main reason why we have preferred geometric reinitialization.

Despite geometric reinitialization has been implemented in parallel, the implementation must still be improved to obtain an acceptable scalability. The results presented in the next Section have been calculated using geometrical reinitialization However, to show the scalability of the method we have calculated 10 time steps without reinitialization on a mesh with 5 Melements (mesh B - next Section). In Figure 2 we can see that the scaling is nearly perfect up to 100 processors, that is 50000 elements per processor.

4. Numerical Results

Numerical results for two different ship hulls are presented. The first one is the David Taylor Model Basin (DTMB) model 5512, a 1:46.6 model scale of a modern

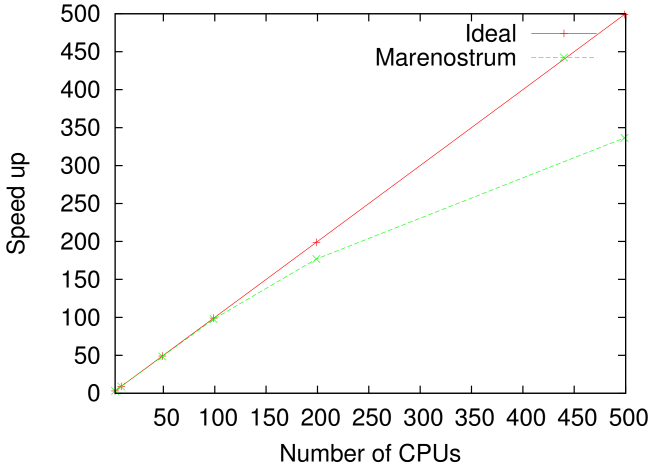

Figure 2: Scalability results up to 500 processors on Marenostrum.

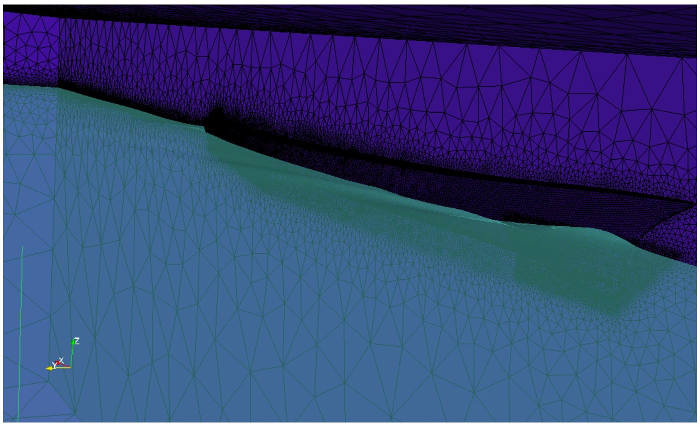

Figure 3: Mesh B and free surface.

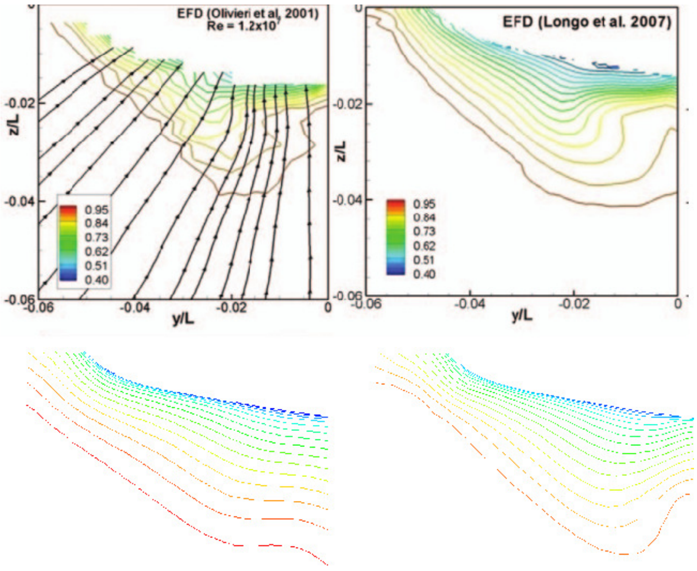

Figure 4: Non-dimensional velocity results at $x/L = 0.95$ for the 5512 hull. Top: experimental data [20]and [21], bottom: numerical results meshes B(left) and C(right).

surface combatant. The second example is a full scale race boat hull.

4.1. DTMB 5512

In order to benchmark our numerical results the flow around the bare hull David Taylor Model Basin (DTMB) model 5512, a 1:46.6 model scale of a modern surface combatant, is used. It has been tested in the towing tanks at DTMB, IIHR (Iowa) [20] and INSEAN (Italy) [21]. It has a sonar dome, which provides additional geometric complexity. The DTMB 5512 model is $L = 3.048\ m$ long with $0.132\ m$ draft. Results at $Re = 4.85 \times 10^6$ and $Fr = 0.28$ will be shown.

Three different finite element meshes have been used. The first one, that we shall call Mesh A, is formed by 8 Melements and 1.5 Mnodes. The second one, Mesh B, is a slightly improved version of the previous one that takes into account symmetry and therefore simulates only half of the whole domain. It is formed by 5 Melements. Finally Mesh C is obtained by dividing mesh B into elements with half the size arriving to a total of 40 Melements. This has been done automatically using the strategy presented in [22]. All three meshes are formed mainly by tetrahedra and include an anisotropic layer of prisms close to the hull. Mesh B is shown in Figure 3 . It is refined close to the ship hull and close to $z = 0$, the region where the free surface is found. As we have said, the other two meshes are quite similar.

In Figure 4 we compare the non-dimensional velocity results at $x/L = 0.95$ we have obtained with meshes B and C against the experimental ones from [20] and [21]. Actually the results from [21] are for the same hull at a bigger scale and Reynolds number but same Froude number and they are usually included in the comparisons (see for example [23]). The results obtained with both meshes reproduce the experimental results satisfactorily. Close to the symmetry face the results with the divided mesh (C) show some improvements compared to the original mesh (B).

Figure 5 presents the non-dimensional turbulent kinetic energy distribution at $x/L = 0.95$. The experimental results are from [20] and the numerical ones from meshes B and C. The discrepancy between numerical an experimental results is higher than for the velocity. Comparable discrepancies can be observed in the literature. For example, in [23], where several numerical and turbulence models have been tested, the discrepancies in the turbulent kinetic energy are even greater than ours and different scales have been used to plot the different numerical and the experimental results. Our numerical results with mesh B are closer to the experimental ones than those obtained with the divided mesh (C). No clear explanation can be found for this behavior and it seems to be only a coincidence.

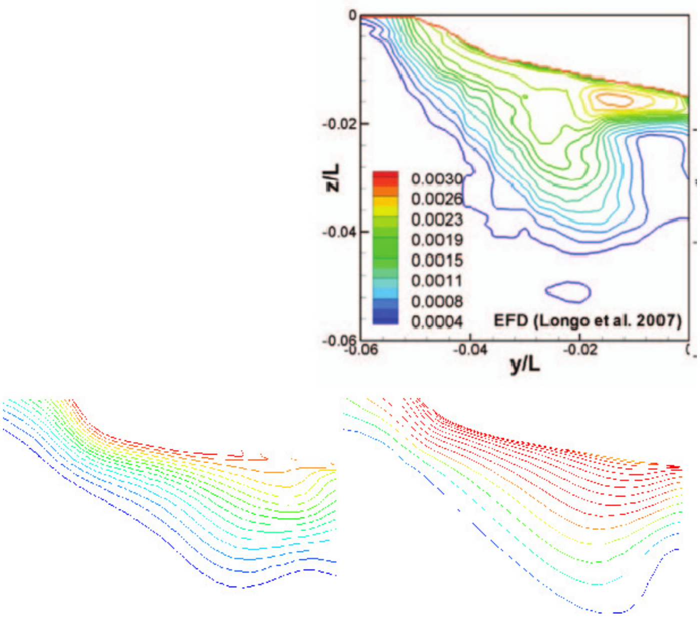

Figure 5: Non-dimensional turbulent kinetic energy distribution at $x/L = 0.95$ for the 5512 hull. Top: experimental data [20], bottom: numerical results meshes B(left) and C(right).

In Figure 6 we compare the wave elevation profiles obtained with meshes B and C against the experimental results from [20]. The result with the original mesh are acceptable but, as one could expect, they are significantly improved with the divided mesh. One wave that is not captured by the original mesh can be observed. It is marked with an arrow in the numerical results corresponding to mesh C in Figure 6.

An additional comparison between the wave elevation profiles behind the ship hull obtained with meshes B and C is presented in Figure 7. It reconfirms the improvements in the shape of the free surface that can be obtained with a finer mesh.

Table 1 presents the forces on the ship hull. The value for the viscous force included in the experimental column is actually not an experimental value but has been obtained from the model-ship correlation line (ITTC 1957). The agreement between numerical and experimental results is very satisfactory. The improvements introduced in mesh B compared to mesh A have had a very positive impact. They have consisted in reducing the element sizes close the bow and stern. Dividing mesh B into elements with half the size has not improved the results for the force on the ship hull. Instead it has worsen the pressure forces slightly. It is important to note that if one is mainly interested on the forces on the hull, very fine meshes do not seem to introduce much improvements, at least with RANS turbulence models.

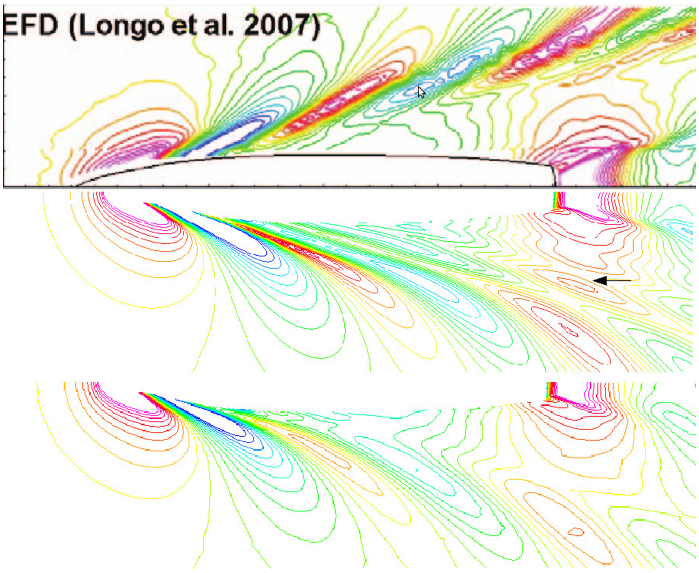

Figure 6: Non dimensional wave elevation profiles for the 5512 hull. Top: experimental data [20], middle: numerical results mesh C, bottom: mesh B. Contours levels are from -5×10^3 to 5×10^3 with intervals of 5×10^4.

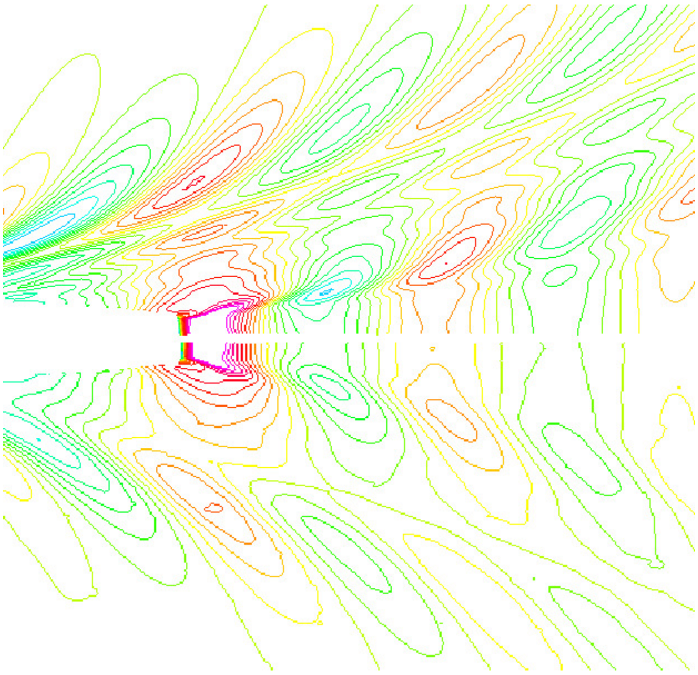

Figure 7: Non dimensional wave elevation profiles for the 5512 hull. Top: mesh C, bottom: mesh B.

	Exp	Mesh A	Mesh B	Mesh C
$F_x[N](total)$	7.432	7.67	7.41	7.30
$F_{Vx}[N]$ (viscous)	5.52	5.68	5.54	5.54
$F_{Px}[N]$(pressure)		1.99	1.87	1.76

Table 1: Forces on the DTMB hull

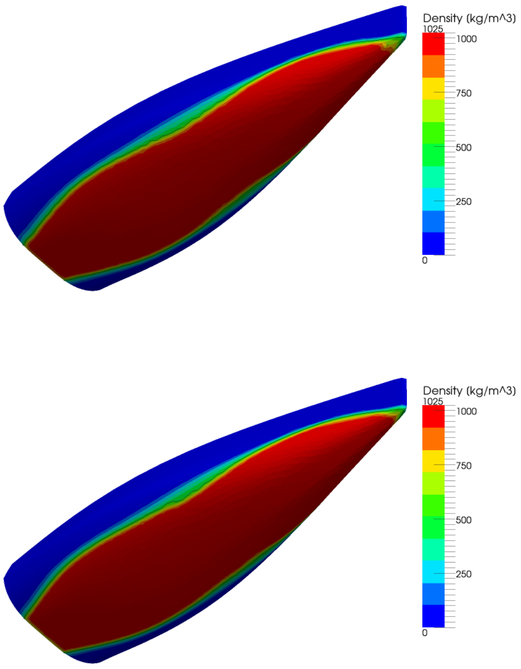

Figure 8: Density on race boat hull. Top: Original mesh, bottom: Divided mesh.

4.2. Full scale race boat hull

The second numerical example involves simulating a real race boat hull. Despite there are no experimental results for this hull it is interesting to test the behavior of our code at a real scale hull that results in a higher Reynolds number flow. It is $L = 10.85m$ long and moves at $5.144m/s$. This result in $Re = UL\rho/\mu = 4.7 * 10^7$ and $Fr = U/\left(\sqrt{gL}\right) = 0.4985$.

Two finite element meshes have been used. The first one is formed by 2.12 Melements and 0.36 Mnodes. The second one is obtained by dividing the original mesh into elements with half the size arriving to total of 16.9 Melements and 2.88 Mnodes. As in the DTMB case, they are formed mainly by tetrahedra and include an anisotropic prisms layer close to the hull.

The density obtained with both meshes is presented in Figure 8. No significant differences can be observed. A region with smoothed properties close to the interface as can be observed as explained in Section 2.

The pressure on the ship hull is presented in Figure 9. Once again very slight differences can be observed. The tangential stress on the ship hull, Figure 9, show only minor differences in the central part of the hull close to the bow.

The excellent agreement observed on the previous three Figures results in a similar agreement for the forces on the race boat hull as shown in Table 2. The differences for total, viscous and pressure forces are less than 2%. This indicates that the method is able to provide very

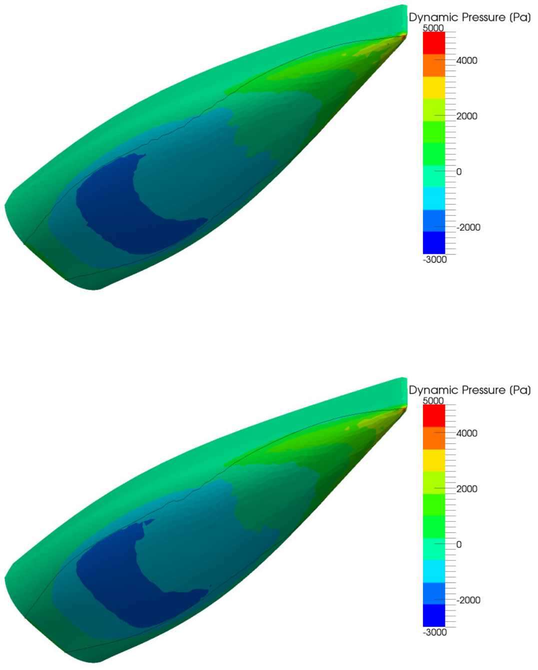

Figure 9: Pressure on race boat hull. Top: Original mesh, bottom: Divided mesh.

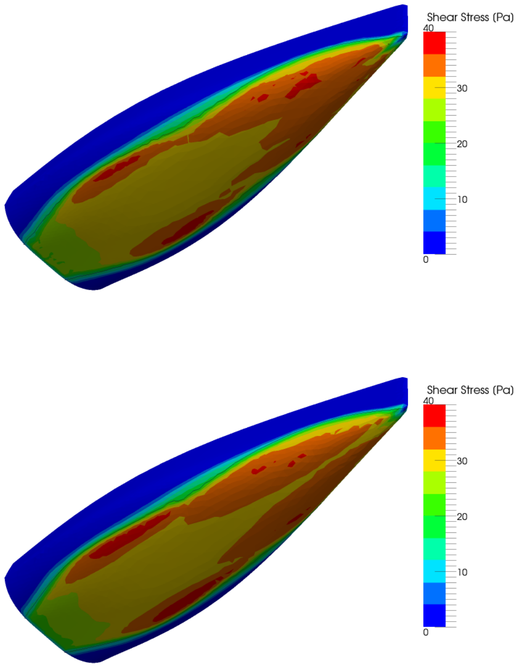

Figure 10: Tangential Stress on race boat hull. Top: Original mesh, bottom: Divided mesh.

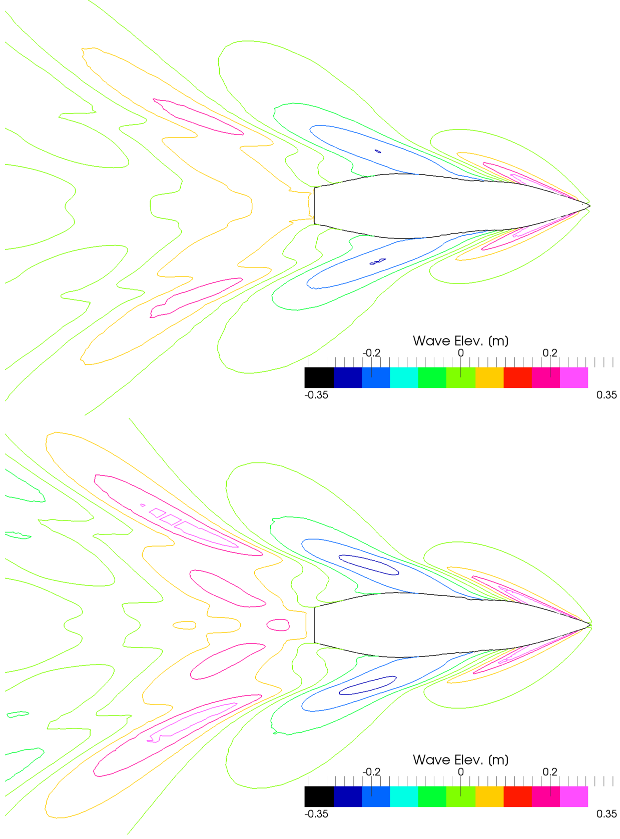

Figure 11: Boat hull Wave Elevation. Top: Original mesh, bottom: Divided mesh.

acceptable results for the forces on the hull with relative coarse mesh, a highly desirable feature.

	Orig. mesh	Div. mesh	Difference
$F_x[N](total)$	2132.0	2161.6	1.37%
$F_{Vx}[N]$ (viscous)	744.0	748.6	0.61%
$F_{Px}[N]$(pressure)	1388.0	1413.0	1.77%

Table 2: Forces on the race boat hull

As in the DTMB example, the wave elevation profile shown in Figure 11 is the result most significantly improved when a fine mesh is used.

5. Conclusions

A two-fluid flow solver has been implemented in the parallel CFD finite element code Alya. It uses a fixed mesh strategy and captures the position of the interface using the Level Set method. Despite the methodology can be applied to a wide range of problems in this work we have focused on Ship Hydrodynamics applications. Comparison with experimental results and application to a full scale race boat hull have shown very promising results even with relatively coarse meshes. Forces on the ship hull can be accurately predicted even with a coarse mesh. To match the wave pattern accurately finer meshes are needed. Turbulent kinetic energy seems to be the most difficult result to calculate, but a similar behavior has been observed in the literature.

The results in this work correspond to the simplest case of a hull in a fixed position without incoming waves. Future work should include incoming waves and a moving hull. Moreover taking into account the high scalability of the code, fleet interaction could be simulated. If improved results are sought, turbulence modelling could be improved restoring to more advanced models such as Detached Eddy Simulation and its improved variants.

References

[1] R. Azcueta and N. Rousselon. CFD applied to super and mega yacht design. In *Design, Construction and Operation of Super and Mega Yachts Conference*, Genova, Italy, 2009. 2009.

[2] S. Osher and R.P. Fedkiw. *Level set methods and dynamic implicit surfaces*. Springer–Verlag, 2003.

[3] J.A. Sethian and P. Smereka. Level set methods for fluid interfaces. *Annu. Rev. Fluid Mech.*, 35:341–372, 2003.

[4] R. Codina. A stabilized finite element method for generalized stationary incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 190:2681–2706, 2001.

[5] P.R. Spalart and S.R. Allmaras. A one-equation turbulence model for aerodynamic flows. *AIAA*, 92-0439, 1992.

[6] F.R. Menter. Zonal two equation k-ω turbulence models for aerodynamic flows. *AIAA*, 93-2906, 1993.

[7] Y. Chang, T. Hou, B. Merriman, and S. Osher. A level set formulation of Eulerian interface capturing methods. *Journal of Computational Physics*, 124:449–464, 1996.

[8] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing soluions to incompressible two-phase flow. *Journal of Computational Physics*, 114:146–159, 1994.

[9] H. Reichardt. Vollständige darstellung der turbulenten geschwindigkeitsverteilung in glatten leitungen. *Zeitschrift fuer Angewandte Mathematik und Mechanik*, 31:208–219, 1951.

[10] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods*. Springer Verlag, 1991.

[11] A.J. Chorin. A numerical method for solving incompressible viscous problems. *Journal of Computational Physics*, 2:12–26, 1967.

[12] R. Temam. Sur l'approximation de la solution des équations de Navier–Stokes par la méthode des pas fractionaires (I). *Archives for Rational Mechanics and Analysis*, 32:135–153, 1969.

[13] J.B. Perot. An analysis of the fractional step method. *Journal of Computational Physics*, 108:51–58, 1993.

[14] A. Quarteroni, F. Saleri, and A. Veneziani. Factorization methods for the numerical approximation of Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 188:505–526, 2000.

[15] J. van Kan. A second-order accurate pressure correction scheme for viscous incompressible flow. *SIAM Journal of Scientific Computing*, 7:870–891, 1986.

[16] P.K.W. Vinsome. Orthomin, an iterative method for solving sparse sets of simultaneous linear equations. In *4th Symposium on reservoir simulations. Society of Petroleum Engineers of AIME*, pages 149–159, 1976.

[17] G. Houzeaux, M. Vázquez, R. Aubry, and J. M. Cela. A massively parallel fractional step solver for incompressible flows. *J. Comput. Phys.*, 228:6316–6332, September 2009.

[18] G. Houzeaux, R. Aubry, and M. Vzquez. Extension of fractional step techniques for incompressible flows: The preconditioned orthomin(1) for the pressure schur complement. *Computers and Fluids*, 44(1):297 – 313, 2011.

[19] A. Khamayseh and A. Kuprat. Deterministic point inclusion methods for computational applications with complex geometry. *Computational Science & Discovery*, 1, 2008.

[20] J. Longo, J. Shao, M. Irvine, and F. Stern. Phase-averaged piv for the nominal wake of a surface ship in regular head waves. *Journal of Fluids Engineering*, 129:524–540, 2007.

[21] A. Olivieri, F. Pistani, A. Avanzini, F. Stern, and R. Penna. Towing tank experiments of resistance, sinkage and trim, boundary layer, wake, and free surface flow around a naval combatant insean 2340 model. Technical Report 421, IIHR Technical Report, 2001.

[22] G. Houzeaux, R. de la Cruz, H. Owen, and M. Vázquez. Parallel uniform mesh multiplication applied to a navier-stokes solver. *Submitted to Computers & Fluids*, 2011.

[23] J. Yang S. Bhushan, P. Carrica and F. Stern. Scalability studies and large grid computations for surface combatant using cfdship-iowa. *The International Journal of High Performance Computing Applications*, published online, 2011.